实验3 简单计算机系统-系统设计A-实验报告

电 25 吴晨聪 2022010311

# 3.1 cpuB模块设计

**(1) 参照上面的步骤，设计、仿真mux2模块.**

两路选择器模块mux2的功能为用1位控制信号来选择某一路输入连接到输出。mux2.v的功能实现比较简单，只需要根据控制信号的内容来进行选择即可，具体测试结果如下:

一張含有 螢幕擷取畫面, 文字, 多媒體軟體, 行 的圖片

AI 產生的內容可能不正確。

图1 mux2模块的测试结果

**(2) 编写顶层文件cpuB.v及其测试文件；通过修改regfile.v，给寄存器赋初值；在测试文件中，设置激励信号，对系统进行仿真验证；分析仿真结果.**

cpuB中需要用到两路选择器的地方主要有三个：一个是寄存器regfile中的nd（即写入单元对应的地址），在R型指令中应该是Rd的值，在I型指令中是Rt，所以要做Rd和Rt之间的选择，控制位信号即为regdes；另一个是alu中第二个运算数字srcB，在R型指令中是寄存器输出的q2，在I型指令中为立即数imm，所以也需要在q2和imm之间做一个选择，控制位信号记为selscrB；此外存入寄存器regfile中的数据di也要用到，在R型指令以及用于计算的I型指令中di都是alu的计算结果s，而在load指令中di的值为RAM输出的值memData，此处也用一个二路选择器来选择s和memData，控制位记为memtoreg。（此外理论上alu的第一位运算位scrA有可能是q1也有可能是PC+1，但后者只会在J型指令中出现，故这里先不考虑）

在设置好对应的二路选择器以后按照实验指导书中的框图搭建cpuB.v（实际的连接图如图2所示），并模拟寄存器地址的输入、运算符的输入、立即数的输入以及各个选择器控制信号的输入可以检测cpuB的运行情况，具体运行结果如图3所示。

一張含有 圖表, 方案, 行, 工程製圖 的圖片

AI 產生的內容可能不正確。

图2 cpuB的连线方式

一張含有 螢幕擷取畫面, 電子產品, 多媒體軟體, 電路 的圖片

AI 產生的內容可能不正確。

图3 cpuB的仿真结果

除了实验指导中的R指令的一些内容以外，还增加了一些I指令以及load指令之类的模拟。运算结果都能和预期相符，由此证明cpuB的搭建逻辑是合理的。

# 3.2 cpuC设计

cpuC的功能需要能从ROM中读取指令并至少能执行R型指令的内容，除了用到之前搭建的模块以外，还需要额外设计一个控制器模块controller.v来根据指令来输出各种控制信号。

控制器controller的搭建逻辑如下。其输入是代码的Op部分，即代码的编号，输出是各种控制信号，这里包括多路选择器的控制信号regdes、selsrcB、memtoreg，寄存器写入允许信号regwrite以及运算符储存器的flagwrite信号。其中由于只涉及R指令，所以regdes、selsrcB、memtoreg和regwrite的值固定为1、0、0、1即可；alucs的值即取Op的后三位；flagwrite则需要根据alus的具体内容来判断，加减运算flagwrite取1，与或选择运算flagwrite取0。

一張含有 圖表, 方案, 行, 工程製圖 的圖片

AI 產生的內容可能不正確。

图4 cpuC的连线方式

在ROM中预先写入实验指导中提供的指令编码，利用addrGen充当程序计数器来不断执行下一条指令，得到的cpuC的仿真运行结果如图5所示。

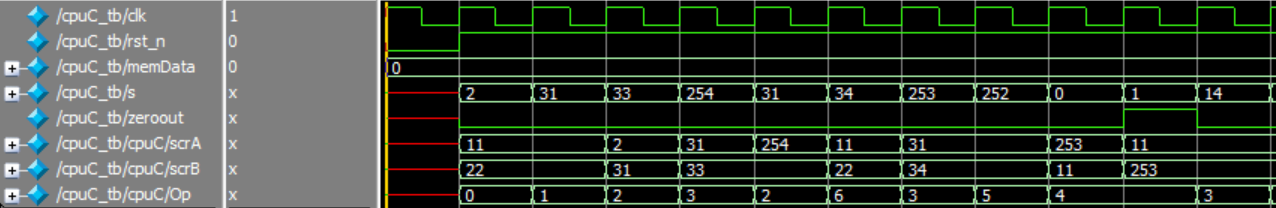


图5 cpuC的仿真结果

该仿真结果符合实际结果，故cpuC的功能正常。

rom存入机器码与汇编语言格式指令之间的关系如下表所示。

|  |  |  |
| --- | --- | --- |
| 行号 | 指令代码 | 机器码 |
| 0 | AND R2,R0,R1 | 0x0180 |
| 1 | OR R3,R0,R1 | 0x11c0 |
| 2 | ADD R2,R2,R3 | 0x2b80 |
| 3 | SUB R3,R3,R2 | 0x3ec0 |
| 4 | ADD R3,R3,R2 | 0x2ec0 |
| 5 | ADDC R2,R0,R1 | 0x6180 |
| 6 | SUB R1,R3,R2 | 0x3e40 |
| 7 | SUBC R2,R3,R2 | 0x5e80 |
| 8 | SLT R2,R1,R0 | 0x4480 |
| 9 | SLT R3,R0,R1 | 0x41c0 |
| 10 | SUB R2,R0,R1 | 0x3180 |
| 11 | SUBC R3,R0,R1 | 0x51c0 |